Ensuring that all Forms Unload

When a Visual Basic program ends, all of its forms should be unloaded and removed from memory. Unfortunately Visual Basic does not take care of this automatically. Particularly if your program contains a lot of forms, it is possible for one or more forms to remain in memory even after the program terminates. Remember, calling a form's Hide method or setting its Visible property to False does not unload it. Even if you explicitly unload a form (using the Unload procedure) it can still take up resources unless the form's reference is set to Nothing. This tip shows you how to ensure that all of a program's forms are unloaded and their resources released upon program termination.

This technique makes use of the fact that a Visual Basic application has a global Forms collection whose elements represent all of the application's loaded forms. You could loop through this collection, unloading all forms as shown in this code snippet:

```
Dim f As Form
   For Each f In Forms
       Unload f
       f = Nothing
   Next f
```

There's a problem with this straightforward approach, however. If you execute this code from a procedure it works fine, but if you call it from the main form's Form_Unload event procedure it will try to unload the main form which is already in the process of unloading itself (otherwise the Form_Unload event procedure would not have fired). You can get around this potential problem as follows:

```
Public Sub UnloadAllForms(Optional FormToIgnore As String = "")

      Dim f As Form
      For Each f In Forms
          If f.Name <> FormToIgnore Then
              Unload f
              f = Nothing
          End If
      Next f

  End Sub
```

With this procedure in place you would call it like this from the main form's Form_Unload event procedure and it will unload all forms except the main form:

```
UnloadAllForms Me.Name
```

If calling it from a separate procedure, pass no argument and the procedure will unload all of the program's forms.

Visual Basic Tips